PLOTIT-METHOD OF INTERACTIVELY PLOTTING
INPUT DATA FOR THE VORLAX COMPUTER PROGRAM

Frederick M. Denn

Vought Corporation
Hampton Technical Center
3221 North Armistead Avenue • Hampton, Virginia 23666
an LTV Company

NASA CONTRACT NAS1-13500
JUNE 1978

NASA Contractor Report 158896

N78-28830

Unclas
G3/61 25903

NASA
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

https://ntrs.nasa.gov/search.jsp?R=19780020887 2019-06-03T15:52:15+00:00Z
SUMMARY

A method of plotting the geometric input to the VORLAX computer program by means of an interactive remote computer terminal is described. The software consists of a procedure file and two programs and was developed for use with the Langley Research Center computer system. The programs and procedure file are described and a sample execution is presented.

INTRODUCTION

The VORLAX computer program uses a sparse set of geometric input data to describe the aircraft configuration being analyzed. The geometry of the configuration can at times become very complex, and it is necessary to plot the configuration resulting from the data in order to ascertain its accuracy. The procedure file and two computer programs described herein provide a method for plotting this data at an interactive graphics terminal. A sample execution of the procedure is presented.

Use of this method allows the configuration to be plotted with any combination of roll, pitch, or yaw angles. Three independent forms of data display are available, and these may be specified in any combination. These are: (1) configurations with or without camber, (2) configurations showing only major panels or only minor panels, and (3) configurations with or without control points plotted. Any section of the plot may be enlarged for examination in greater detail.

The procedure file and computer programs to plot the VORLAX input data have been written to be used in the Langley Research Center computer system which provides a Network Operating System (NOS) and a Tektronix Plot 10 package. Langley Research Center users will find the procedure file and computer programs in mass storage as public files in the catalog of user number 214737C.
DISCUSSION

DESCRIPTION OF PROCEDURE FILES AND PROGRAMS

A method has been developed for plotting the VORLAX input data which consists of a procedure file and two computer programs. The procedure file PLOTIT and the computer programs READS and PLOTS are described in the following. A sample execution is provided which includes illustrations of the displays.

Procedure File PLOTIT

Procedure file PLOTIT (Appendix A) is used to simplify the plotting procedure. PLOTIT first gets the desired data file and renames it TAPE1. It then gets the binary form of program READS, which reads the input data and prepares files suitable for the subsequent operations. PLOTIT then gets and executes the binary form of PLOTS, which performs the aircraft geometry plotting.

Program READS

Program READS (Appendix B) reads the VORLAX input data from a disc file named TAPE1 and determines the necessary scaling factor in order for the aircraft drawing to fit on the screen.

The data for each of the panels are then read, and the coordinates of a set of points that describe the panels are calculated. These coordinates are stored on three disc files named TAPE3, TAPE4, and TAPE5. These data files will be read by program PLOTS.

Program READS provides the user with two methods of representing the aircraft. The first method displays the aircraft with camber as shown in figure 1(a). The second method shows the aircraft without camber. Figure 1(b) is the aircraft of figure 1(a) without camber.

Program PLOTS

Program PLOTS (Appendix C) reads the data produced by program READS. This program offers the user several variations in the display which are described
in the following sections.

**Rotation of the Aircraft.** - The initial position of the aircraft is a side view with the nose to the left. From this position, it is possible to rotate the aircraft first in roll, then in pitch, and finally in yaw. Positive angles are defined as follows: roll, right wing down; pitch, nose up; and yaw, nose right.

**Additional Independent Variations.** - PLOTS offers two additional independent variations in the display. These variations are: (1) with or without subpaneling and (2) with or without control points. Examples of these variations are shown in figures 1(c) through 1(f). Figure 1(c) shows the aircraft without subpaneling or control points; figure 1(d) has subpaneling added; in figure 1(e) control points only have been added; and in figure 1(f) control points and subpaneling have been added. All of the plots are constructed using orthographic projection.

**Recovery from Input Errors.** - If an error has been made in specifying the input parameters, the execution of PLOTS can be stopped. There are three ways to stop PLOTS while it is executing. These are: (1) enter a value greater than 360 for roll angle, (2) answer any of the questions with STOP, or (3) stop the program while it is plotting by interrupting it with the break key and then entering an S.

Program PLOTS can be restarted at the beginning by sending the command PLOTB to the computer. PLOTB will also restart the program if it is stopped for any other reason.

**SAMPLE EXECUTION**

Figure 2 shows a sample execution of PLOTIT at a remote terminal. The first command gets the procedure file PLOTIT.

GET,PLOTIT/UN=214737C

The second command initiates the execution of the procedure file.
CALL,PLOTIT(T=VORLAXX)

VORLAXX is the name of the file on which the VORLAX data deck has been stored for this example. The file name in the calling statement can be any name which corresponds to a file on which VORLAX data is stored.

The first question asked by the computer deals with camber in the panels.

DO YOU WANT CAMBER IN THE PANELS?
TRUE OR FALSE

If camber is desired in the panels, type in TRUE, otherwise type in FALSE and the camber will be set equal to zero.

The next three questions asked by the computer are concerned with the desired roll, pitch, and yaw angles of the configuration, and are as follows.

INPUT THE ROLL ANGLE FOR THE AIRCRAFT (DEG), > 360 TO STOP.
PITCH ANGLE
YAW ANGLE

The angles desired in degrees, are typed in after the questions. If termination of the program is desired, a value greater than 360 may be typed in for the roll angle.

The next two questions are concerned with the desirability of displaying paneling and control points. The two questions are:

SUBPANELING ? TRUE OR FALSE
CONTROL POINTS ? TRUE OR FALSE

If these questions are answered TRUE, the subpaneling and control points are incorporated into the plots. If they are answered FALSE, then these quantities are deleted. The plot resulting from the input in figure 2 is presented in figure 3.

If a certain section of the plot needs to be enlarged in order to examine the plotted data more closely, this may be accomplished at the terminal. When the computer has finished plotting, it will print the following statement:

FOR ENLARGEMENT INPUT YES
At this point a hard copy can be made if desired. If any reply except YES is typed in, the computer will ask for a new set of angles. If YES is typed in, the graphics cursor (cross hairs) will appear. The cursor should then be located at the lower left corner [fig. 4(a)] of a rectangular region to be enlarged. A non-control keyboard character should be pressed. This will cause the cursor to disappear. The carriage return is then pressed. This sequence sends the coordinates of the first corner to the computer. The graphics cursor will reappear and should be relocated to the upper right corner of the desired rectangular region [fig. 4(b)]. A non-control keyboard character and the carriage return are then pressed as for the first corner. An enlargement of the region defined by these positions of the graphics cursor is shown in figure 4(c).

When the plot is finished, the computer will again print

```
FOR ENLARGEMENT INPUT YES
```

This allows a further enlargement of a section of the plot if desired.

**CONCLUDING REMARKS**

A plotting routine, PLOTIT, has been developed for plotting the input data for the VORLAX computer program. This program allows the user to plot geometric input data interactively at a remote graphics terminal and thereby ascertain very rapidly whether or not the data is correct.

The routine consists of two programs and a procedure file. These have been designed for use on the Control Data Corporation computer system with a Network Operating System (NOS) and a Tektronix Plot 10 graphics package at the NASA Langley Research Center.
REFERENCES

APPENDIX A

PROCEDURE FILE PLOTIT

This procedure file gets the binary form of READS(READB) and executes it, then gets the binary form of PLOTS (PLOTB) and executes it.

PLOTIT.
RETURN,TAPE3,TAPE4,TAPES.
GET,TAPE1=T.
GET,READB/UN=214737C.
READB.
RETURN,READB.
GET,PLOTB/UN=214737C.
PLOTB.
EXIT.
APPENDIX B
SOURCE LISTING OF PROGRAM READS

PROGRAM READS(INPUT, OUTPUT, TAPE1, TAPE4, TAPE2=OUTPUT, TAPE3, TAPE5) READS 1
THIS PROGRAM READS THE INPUT DATA FOR THE VORLAX PROGRAM READS 2
SO IT CAN BE PLOTTED.
READS 3

DIMENSION TITLE(8) READS 4
COMMON/BLOCK/XOFFSET READS 5
LOGICAL CAMBER READS 6
REAL LENGTH READS 7
READ(1,100)TITLE READS 8
READ(1,103)LAX,LAY READS 9
READ(1,101) READS 10
READ(1,102)NPAN,WSPAN READS 11
CALL SIZES(NPAN,LENGTH,XOFFSET,WSPAN) READS 12
IF(WSPAN.GT.WSPAN)WSPAN=WSPAN READS 13
TESTRTO=LENGTH/WSPAN READS 14
IF(LENGTH.GT.WSPAN)WSPAN=LENGTH READS 15
REALY IS THE WIDTH OF THE PLOTTING SURFACE. READS 16
WRITE(3)NPAN,TITLE,TESTRTO READS 17
REALY=10.0 READS 18
REALY=REALY-.20 READS 19
SCALE=REALY/WSPAN READS 20
XOFFSET=-(XOFFSET+WSPAN*.5) READS 21
WRITE(2,104) READS 22
READ 105,CAMBER READS 23
DO 200 I=1,NPAN READS 24
200 CALL PANLRED(SCALE,LAX,LAY,CAMBER) READS 25
100 FORMAT(8A10) READS 26
101 FORMAT(//) READS 27
102 FORMAT(12,48X,F10.0) READS 28
103 FORMAT(11X,11X,F11.1) READS 29
104 FORMAT("DO YOU WANT CAMBER IN THE PANELS ? ",/,"TRUE OR FALSE") READS 30
105 FORMAT(L7) STOP READS 31
END READS 32
READS 33
SUBROUTINE PANELRED(SCALE, LAX, LAY, CAMBER)
COMMON BLOCK/XOFFSET
COMMON BLOKO/NAP, XAF(50), ZC(2,50), CORDI, CORD2, CD1SIN, CD2SIN
COMMON/CANDS/SIN1, SIN2, COS1, COS2, DELTAY
COMMON/R0J1/R0J, CROSSIZ
LOGICAL CAMBER
INTEGER PDC, RNC

C THIS SUBROUTINE READS THE VORLAX DATA FOR A PANEL EACH TIME IT
C IS CALLED. THE COORDINATES OF A SET OF POINTS THAT DESCRIBE
C THE PANEL ARE CALCULATED AND STORED ON DISC.

DIMENSION X(2), Y(2), Z(2), CORD(2), RLE(2)
COMMON PHI0, PHI(100), R0(100), SINE(100), COSINE(100)
COMMON VORS(3,500)
COMMON/PI/PIE
COMMON/TWIST/AINC1, DAINC
REAL K
DATA PIE/3.14159/, CROSSIZ/.02/

C THIS SECTION READS THE VORLAX DATA CARDS FOR A PANEL.

DO 200 I=1,2
200 READ(1,100)X(I), Y(I), Z(I), CORD(I)
DO 250 I=1,2
X(I) = (X(I) + XOFFSET) * SCALE
Y(I) = Y(I) * SCALE
Z(I) = Z(I) * SCALE
250 CORD(I) = CORD(I) * SCALE
READ(1,101)TV0R, TNVC, PDL
RNCV = INT(TNVC)
NVOR = INT(TV0R)
NVOR1 = NVOR + 1
IF (PDL.LE.360.) GO TO 1
READ(1,102)(PHI(N), R0(N), N=1, NVOR1)
DO 251 N=1, NVOR1
PHI(N) = PHI(N) * 3.14159/180.
251 R0(N) = R0(N) * SCALE
1 READ(1,103)AINC1,AINC2,ITS,NAP,IQUANT,ISYNT,NPP
   AINC1=ATAN(AINC1)
   AINC2=ATAN(AINC2)
   DAINC=AINC2-AINC1
   IF(IQUANT.EQ.0)IQUANT=2
   IF(ISYNT.NE.0)READ(1,104)
   IF(NAP.LE.2)GO TO 2
   READ(1,102)(XAF(I),I=1,NAP)
   DO 88 I=1,NAP
   88 XAF(I)=XAF(I)*.01
      IF(ITS.EQ.0.OR.PDL.GE.360.)GO TO 3
      READ(1,105)RLE(1)
      RLE(1)=RLE(1)*SCALE
      READ(1,102)(ZC(I),I=1,NAP)
      IF(ITS.EQ.0.OR.PDL.GT.360.)GO TO 4
      READ(1,105)RLE(2)
      RLE(2)=RLE(2)*SCALE
      READ(1,102)(ZC(I),I=1,NAP)
      IF(.NOT.CAMBER.A.NAP.GT.2)CALL ZEROZC1(PDL)
   CONTINUE
100 FORMAT(4F10.0)
101 FORMAT(2F10.0,10X,F10.0)
102 FORMAT(8F10.0)
103 FORMAT(2F10.0,2X,9X,II,9X,II)
104 FORMAT(7)
105 FORMAT(F10.0)
C
   THIS SECTION CALCULATES THE LOCATION OF A SET OF POINTS
   THAT DEFINE A FLAT PANEL.
C
   NVRN=(RNCV)*(NVOR+1)+1
   NVRNPRD=(RNCV+2)*(NVOR+1)
   N4= 4.0*RNCV
   N2=2.0*RNCV
   X1=X(1)
   CORD1=CORD(1)
C***** IF THE PANEL IS CURVED GO TO 5
   IF(PDL.GT.360.)GO TO 5
   X2=X(2)
   READS  72
   READS  73
   READS  74
   READS  75
   READS  76
   READS  77
   READS  78
   READS  79
   READS  80
   READS  81
   READS  82
   READS  83
   READS  84
   READS  85
   READS  86
   READS  87
   READS  88
   READS  89
   READS  90
   READS  91
   READS  92
   READS  93
   READS  94
   READS  95
   READS  96
   READS  97
   READS  98
   READS  99
   READS 100
   READS 101
   READS 102
   READS 103
   READS 104
   READS 105
   READS 106
   READS 107
   READS 108
   READS 109
   READS 110
Y1=Y(1)
CORD=CORD(2)
SIN1=SIN(AINC1)
SIN2=SIN(AINC2)
COS1=COST(AINC1)
COS2=COST(AINC2)
CD1SIN=CORD1*SN1
CD2COS=CORD1*COS1
CD2SIN=CORD2*SN2
CD2COS=CORD2*COS2
Z1=Z(1)
Z2=Z(2)
DELTA=Y(2)-Y1
DELAx=(X2-X1)
DELAZ=(Z2-Z1)
DELTA0=(CD2COS-CO1COS)
YRAT=SRT(DELTA*DELAZ+DELTAY*DELTAY)

C
ISETS=ISETS(NVNRPRD)
CALL SAVE(IFUG,3,1DUM,1)
WRITE(3) NVOR, NCVR, POL, IQUANT, NVNRPRD, ISETS
CALL INTERP(0.0, Z0, Z, X0, XERP)
DO 310 I=1, NVOR
IF(LAY.EQ.1) RATIO=(I-1.)/NVOR
IF(LAY.EQ.0) RATIO=.5*(1.-COS(PIE*(I-1.)/NVOR))
VORS(1, I)=X1*X0+RATIO*(DELA+XERP)
VORS(2, I)=Y1+DELTAY*RATIO
VORS(3, I)=Z0+RATIO*(Z2*DELTAY)+Z1

C
DO 501 J=1, NVOR
IF(LAY.EQ.1) PCORD=(I/NVOR)*4.0-3.0/N4
IF(LAY.EQ.0) PCORD=.5*(1.-COS((2.*I/NVOR-1.)*PIE/N2))
CALL INTERP(PCORD, Z0, Z, X0, XERP)
DO 501 J=1, NVOR
I2=I+J*IFUG
IF(LAY.EQ.1) RATIO=(J-1.)/NVOR
IF(LAY.EQ.0) RATIO=.5*(1.-COS(PIE*(J-1.)/NVOR))
VORS(2,I2) = Y1 + DELTAY * RATIO
VORS(3,I2) = Z1 + DELTAZ * RATIO
XCORD = CDICOS * (DELTAC0) * RATIO
XX = X1 * RATIO * (DELTAX)
VORS(3,I2) = VORS(1,I2) + Z20 + ZZ * RATIO
VORS(1,I2) = XX + PCORD + XCORD + X00 * RATIO * XERP

501 IF (I2 .EQ. 500) CALL SAVE(IFUG,3,IFDUM,2)
CALL INTERP(1.0,ZZ01,ZZ,X001,XERP1)
DO 300 I = 1,NVOR1
I2 = NVK1 + NVOR1 + I - 1 + IFUG
IF (LAY .EQ. 1) RATIO = (I = 1) / NVOR
IF (LAY .EQ. 0) RATIO = 5 * (1.0 - COS(PIE * (I - 1.0) / NVOR))
VORS(2,I2) = Y1 + DELTAY * RATIO
VORS(3,I2) = ZZ01 + RATIO * (ZZ1 + DELTAZ) + Z1
VORS(1,I2) = (DELTAX + DELTAC0 + XERP1) * RATIO + X1 + CDICOS + X001

300 IF (I2 .EQ. 500) CALL SAVE(IFUG,3,IFDUM,2)
CALL SAVE(IFUG,3,NNVNRPR0,3)

C * * * * * * * * * * * * * * * * * * * * *
C
C THIS SECTION OF THE PROGRAM CALCULATES THE LOCATION OF THE
C CONTROL POINTS FOR A FLAT PANEL SUCH AS A WING. IT THEN
C PLACES AN "X" ON EACH POINT.
C
PRD = NVOR * RNCV
NNO = (RNCV-1) * NVOR + 1
NNN = PRD + 4
ISET = ISETSF(NNN)
WRITE(4,NVOR,RNCV,PRD,NNN,NNN,NNO,ISET)
CALL SAVE(IFUG,4,IFDUM,1)
DO 1001 I = 1,NNO,NVOR
K = (I - 1) / NVOR + 1
IF (LAX .EQ. 0) PCORD = 5 * (1.0 - COS(K * PIE / RNCV))
IF (LAX .EQ. 1) PCORD = (4.0 * K - 1.0) / 4
CALL INTERP(PCORD,ZZ01,ZZ,X001,XERP)
YRAT = SQRT((DELTAY * DELTAY + (DELTAZ + ZZ) ** 2.0))
VI = CROSSZ * DELTAY / YRAT
V2 = CROSSZ * (DELTAZ + ZZ) / YRAT
DO 1001 J = 1,NVOR

reads 148
reads 149
reads 150
reads 151
reads 152
reads 153
reads 154
reads 155
reads 156
reads 157
reads 158
reads 159
reads 160
reads 161
reads 162
reads 163
reads 164
reads 165
reads 166
reads 167
reads 168
reads 169
reads 170
reads 171
reads 172
reads 173
reads 174
reads 175
reads 176
reads 177
reads 178
reads 179
reads 180
reads 181
reads 182
reads 183
reads 184
reads 185
J2 = (I + J - 2) * 4 + IFUG
J21 = J2 + 1
J22 = J2 + 2
J23 = J2 + 3
J24 = J2 + 4
IF (LAYER.EQ.1) RATIO = (J - 5) / NVOR
IF (LAYER.EQ.0) RATIO = 25 * (2 - COS(P*J/NVOR) - COS(P*(J - 1)/NVOR))
CALL ANGLE (RATIO, WS, WC)
XCORD = COS * DELTA * RATIO
TEMX = X1 + RATIO * (DELTAX + XERP) + PCORD * XCORD + X00
VORS(1, J21) = TEMX - CROSSIZ * WC
VORS(1, J22) = TEMX + CROSSIZ * WC
VORS(1, J23) = VORS(1, J24) = TEMX
TEMY = Y1 + DELTAY * RATIO
VORS(2, J21) = VORS(2, J22) = TEMY
TEMZ = Z1 + DELTAZ * RATIO
TEMZ = TEMZ + ZZG + ZZ * RATIO
VORS(3, J21) = TEMZ - CROSSIZ * WS
VORS(3, J22) = TEMZ + CROSSIZ * WS
VORS(3, J23) = TEMZ - V2
VORS(3, J24) = TEMZ + V2
VORS(2, J23) = TEMY - V1
VORS(2, J24) = TEMY + V1
IF (J24.EQ.500) CALL SAVE(IFUG, 4, IDUM, 2)
1001 CONTINUE
CALL SAVE(IFUG, 4, NNN, 3)
1010 CONTINUE
RETURN

** THIS SECTION OF THE PROGRAM IS FOR CURVED MAJOR PANELS. **

YNOT = Y(I) - RO(I) * COS(PHI(I))
ZNOT = Z(I) - RO(I) * SIN(PHI(I))
TEMP2 = X1 + CORD1

** THIS CALCULATES THE SCALING FACTORS FOR THE RADIUS VECTORS FROM AREA RATIOS. **

IF (NAP.LE.2) GO TO 90
DO 44 J = 1, NAP
APPENDIX B - Continued

89  ZC(C,J) = SQRT(ZC(2,J)*.01)
C
C THIS SECTION CALCULATES THE LOCATION OF A SET OF POINTS THAT
C DESCRIBE A CURVED MAJOR PANEL SUCH AS A FUSELAGE OR A NACELLE
C WITH SUBPANELING.
C
90 CALL INTERP2(1.0,XSHIFT1,SKAL1,DM,DM,ZSHIFT1,DM,9H NO POINT)
CALL SAVE(IFUG,5,IDUM,1)
ISETS=ISETS+(NVNRPD)
WRITE(5,NVOR,RNCV,NVNRPD,ISETS)
DO 1333 I=1,NVOR1
R=RO(I)
FI=PHI(I)
VORS(1,I)=X1
VORS(2,I)=R*COS(FI)*SKAL1+YNOT
VORS(3,I)=R*SIN(FI)*SKAL1+ZNOT+ZSHIFT1
C
DO 1501 J=1,NVOR1
IF(LAX.EQ.1)PCORD=(I/NVOR1*4,-3.)/N4
IF(LAX.EQ.0)PCORD=5*(1.+COS((2.*I/NVOR1-1.)*PIE/N2))
CALL INTERP2(PCORD,XSHIFT,SKAL,DM,DM,ZSHIFT,DM,9H NO POINT)
XTEMP=X1*XSHIFT
DO 1501 J=1,NVOR1
I2=I+J+IFUG
R=RO(I) J)*SKAL
FI=PHI(J)
VORS(1,I2)=XTEMP
VORS(2,I2)=R*COS(FI)*YNOT
VORS(3,I2)=R*SIN(FI)*ZNOT+ZSHIFT
1501 IF(I2.EQ.500)CALL SAVE(IFUG,5,IDUM,2)
CALL INTERP2(1.0,XSHIFT2,SKAL2,DM,DM,ZSHIFT2,DM,9H NO POINT)
DO 1300 I=1,NVOR1
I2=NVRN+NVOR1+I-1+IFUG
R=RO(I)
FI=PHI(I)
VORS(1,I2)=T-MP2
VORS(2,I2)=R*SKAL2*COS(FI)*YNOT
VORS(3,I2)=R*SKAL2*SIN(FI)*ZNOT+ZSHIFT2

READS 224
READS 225
READS 226
READS 227
READS 228
READS 229
READS 230
READS 231
READS 232
READS 233
READS 234
READS 235
READS 236
READS 237
READS 238
READS 239
READS 240
READS 241
READS 242
READS 243
READS 244
READS 245
READS 246
READS 247
READS 248
READS 249
READS 250
READS 251
READS 252
READS 253
READS 254
READS 255
READS 256
READS 257
READS 258
READS 259
READS 260
READS 261
IF (I2.EQ.500) CALL SAVE (IFUG, 5, IDUM, 2)
CALL SAVE (IFUG, 5, NVRNPRD, 3)

THIS SECTION CALCULATES THE LOCATION OF A SET OF POINTS THAT
DESCRIBE A CURVED MAJOR PANEL, WITHOUT SUBPANELING.
NVRN = (NAP) * (NVOR + 1)
NVRNPRD = (NAP) * (NVOR + 1)
ISETS = ISETS (NVRNPRD)
WRITE (3) NVOR, NAP, POL, IQUANT, NVRNPRD, ISETS
CALL SAVE (IFUG, 3, NVRNPRD, 1)
IF (NAP.LE.2) GO TO 556
DO 555 I = 1, NVRN, NVOR1
II = (I) / NVOR1 + 1
XTEMP = X1 + CORI1 * XAF (II)
SKAL = ZC (2, II)
DO 555 J = 1, NVOR1
R = RO (J) * SKAL
FI = PHI (J)
I2 = I + J - 1 + IFUG
VORS (1, I2) = XTEMP
VORS (2, I2) = R * COS (FI) + YNOT
VORS (3, I2) = R * SIN (FI) + ZNOT + RO (J) * COS (PHI (J))

IF (I2.EQ.500) CALL SAVE (IFUG, 3, IDUM, 2)
IF (NAP.GT.2) GO TO 5555

NVRNPRD = NAP * NVOP1
DO 5554 J = 1, NVOR1
J2 = J + NVOR1
VORS (1, J) = X1
VORS (1, J2) = X1 + CORI1
VORS (2, J) = VORS (2, J2) + YNGT * RO (J) * COS (PHI (J))
VORS (3, J) = VORS (3, J2) + ZNOT + RO (J) * SIN (PHI (J))

CALL SAVE (IFUG, 3, NVRNPRD, 3)

THIS SECTION CALCULATES THE LOCATION OF THE CONTROL POINTS FOR
A CURVED MAJOR PANEL. IT THEN PLACES AN "X" ON EACH POINT.
POD = PO (1)
DO 2010 N = 1, NVOR

READS 262
READS 263
READS 264
READS 265
READS 266
READS 267
READS 268
READS 269
READS 270
READS 271
READS 272
READS 273
READS 274
READS 275
READS 276
READS 277
READS 278
READS 279
READS 280
READS 281
READS 282
READS 283
READS 284
READS 285
READS 286
READS 287
READS 288
READS 289
READS 290
READS 291
READS 292
READS 293
READS 294
READS 295
READS 296
READS 297
READS 298
READS 299
\( AZO = RO(N) \cdot \sin(\phi(N)) \)
\( AZL = RO(N+1) \cdot \sin(\phi(N+1)) \)
\( AYO = RO(N) \cdot \cos(\phi(N)) \)
\( AYL = RO(N+1) \cdot \cos(\phi(N+1)) \)
\( DAZ = AZO - AY \)
\( DAY = AYO - AY \)
\( H = \sqrt{OAZ \cdot DAZ + DAY \cdot DAY} \)
\( \sin(N) = DAZ / H \)
\( \cos(N) = AYO / H \)
\( R1 = RO(N) \)
\( R2 = RO(N+1) \)
\( F1 = \phi(N) \)
\( F2 = \phi(N+1) \)
\( RO(N) = (R1 \cdot \sin(F1) + R2 \cdot \sin(F2)) \cdot 0.5 \)
\( \phi(N) = (R1 \cdot \cos(F1) + R2 \cdot \cos(F2)) \cdot 0.5 \)
\( PRD = NVOR \cdot RNCV \)
\( NNO = (RNCV - 1) \cdot NVOR + 1 \)
\( NNN = PRD \cdot 4 \)
\( ISETS = ISETSF(NNV) \)
\( WRITE(4,NVOR,RNCV,PRD,NNN,NNO,NNNO,NNN) \)
\( CALL SAVE(IFUG,IOUH,1) \)
\( DO 2001 I=1,NVG,NVOR \)
\( K = (I-1) / NVOR + 1 \)
\( IF(LAX.EQ.0) PCORD = 5 \cdot (1. - \cos(K \cdot \Pi / RNCV)) \)
\( IF(LAX.EQ.1) PCORD = (4. \cdot \Pi - 1.) / N4 \)
\( CALL INTERP2(PCORD,TEMX,SKAL,DSKL,CROSS,CAMB,DZ,6HPINTS) \)
\( TEMX = TEMX \cdot X1 \)
\( DO 2001 J=1,NVOR \)
\( J2 = (1+J-2) \cdot 4 + IFUG \)
\( J21 = J2 + 1 \)
\( J22 = J2 + 2 \)
\( J23 = J2 + 3 \)
\( J24 = J2 + 4 \)
\( VS = \sin(E(J) \cdot CROSSIZ \)
\( HS = \cos(E(J) \cdot CROSSIZ \)
\( TEMY = \phi(J) \cdot SKAL + YNOT \)
\( TY2 = DSKL \cdot \phi(J) \)
\( TEMZ = RO(J) \cdot SKAL + ZNOT + CAMB \)
APPENDIX B. - Continued

SUBROUTINE INTERP(P,ZA,Z,XA,X)

C THIS SUBROUTINE DOES THE INTERPOLATION FOR CAMBER OF A FLAT SECTION.

C MAJOR PANEL.

COMMON/BLKNO/NAP,XAF(50),ZC(2,50),CORD1,CORD2,CD1SIN,CD2SIN
COMMON/CANDS/SIN1,SIN2,COS1,COS2,DELTA,Y
COMMON/TWIST/AINC,DAINC
COMMON/SLOPE/ANGL,DANG

IF(NAP.LE.2.OR.DELTAY.EQ.0.0)GO TO 3

DO 1 I=2,NAP
1 IF(P.LE.XAF(I))GO TO 2

2 DAF=XAF(I)-XAF(I-1)

DZC1=ZC(I,1)-ZC(I-1,1)
DZC2=ZC(I,2)-ZC(I-1,2)
A1=ATAN(DZC1/DAF*0.1)
A2=ATAN(DZC2/DAF*0.1)
DAA=A2-A1
PC=(P-XAF(I-1))/DAF
CAMS1=(DZC1*PC+ZC(I,1,1-1))*CORD1+C1

READS 338
READS 339
READS 340
READS 341
READS 342
READS 343
READS 344
READS 345
READS 346
READS 347
READS 348
READS 349
READS 350
READS 351
READS 352
READS 353
READS 354
READS 355
READS 356
READS 357
READS 358
READS 359
READS 360
READS 361
READS 362
READS 363
READS 364
READS 365
READS 366
READS 367
READS 368
READS 369
READS 370
READS 371
READS 372
READS 373
SUBROUTINE INTERP2(P,X,S,DS,CROS,Z,DZ,TEST)
THIS SUBROUTINE DOES THE INTERPOLATION FOR THE RADIUS VECTORS
AND THE CAMBER OF A CURVED MAJOR PANEL.
COMMON/BLOKO/NAP,XAF(50),ZC(2,50),CORD1,CORD2,CD15IN,CD25IN
COMMON/ROJ1/ROJ,CROSSIZ
X=P*CORD1
IF(NAP.LE.2)GO TO 3
DO 1 I=2,NAP
1 IF(P.LE.XAF(I))GO TO 2
2 DAF=XAF(I)-XAF(I-1)
DZC1=ZC(1,I)-ZC(1,I-1)
DZC2=(ZC(2,I)-ZC(2,I-1))
P=p-XAF(I-1))/DAF
S=ZC(2,I-1)+DZC2*PC
Z=ZC(1,I-1)+DZC1*PC
Z=Z01*CORD1
IF(TEST.NE.6HPINTS)RETURN
DRAD=ROJ*DZC2
DELX=DAF*CORD1
COSINE=DELX/SRT(DELX*DELX)
CROS=COSINE*CROSSIZ
P2=CROS/CORD1
READS 388
READS 389
READS 390
READS 391
READS 392
READS 393
READS 394
READS 395
READS 396
READS 397
READS 398
READS 399
READS 400
READS 401
READS 402
READS 403
READS 404
READS 405
READS 406
READS 407
READS 408
READS 409
PC2 = (P + P2 - XAF(I-1))/DAF
S2 = ZC(2, I-1) + DZC2*PC2
Z2 = ZC(1, I-1) + DZC1*PC2
DZ = Z2 - Z
DS = S2 - S
RETURN
S = S2 = 1.0
Z = DS = DZ = 0.0
CROS = CROSSIZ
RETURN
END

SUBROUTINE ANGLE(R, S, C)
C THIS SUBROUTINE IS USED FOR PLACING X'S ON CONTROL POINTS
C ON FLAT MAJOR PANELS
COMMON/SLOPE/ANGLE, DANG
ANG = ANGL + R*DANG
S = SIN(ANG)
C = COS(ANG)
RETURN
END

SUBROUTINE ZEROZC1(DPL)
C THIS SUBROUTINE REMOVES CAMBER FROM THE MAJOR PANELS
COMMON/BLK0/NAK, XAF(50), ZC(2, 50), CORD1, CORD2, CD1SIN, CD2SIN
DO 1 I = 1, 2
IF(I.EQ.2, A, DPL.GE.360.) RETURN
DO 1 J = 1, NAP
1 ZC(I,J) = 0.0
RETURN
END
FUNCTION ISETSF(N)  
THIS FUNCTION DETERMINES THE NUMBER OF DATA SETS NEEDED TO  
DESCRIBE A MAJOR PANNEL.  
R=1.0/500.*N  
ISETSF=INT(R)  
IF(ISETSF+LT.R)ISETSF+ISETSF+1  
RETURN  
END

SUBROUTINE SAVE(FUG,FILE,PRD,TEST)  
THIS SUBROUTINE MAKES IT POSSIBLE TO REDUCE CORE REQUIREMENTS  
BY REDUCING ARRAY SIZE.  
INTEGER FUG,FILE,PRD,TEST,PRD  
COMMON VORS(3,500)  
IF(TEST.EQ.2)GO TO 2  
IF(TEST.EQ.3)GO TO 3  
FUG=0  
RETURN  
2  
FUG=FUG-500  
WRITE(FILE,:)((VORS(I,J),I=1,3),J=1,500)  
RETURN  
3  
PNTS=MOD(500,500)  
IF(PNTS.EQ.0)RETURN  
WRITE(FILE,:)((VORS(I,J),I=1,3),J=1,PNTS)  
RETURN  
END

SUBROUTINE SIZES(NPAN,LENGTH,XOFFSET,WIDTH)  
** DETERMINES THE LENGTH, WIDTH, STARTING POINT OF THE  
** AIRCRAFT.  
REAL LENGTH  
NPAN2=NPAN-1  
CALL FIND(LENGTH,XOFFSET,WIDTH)  
IF(NPAN.EQ.1)GO TO 8
SUBROUTINE FIND
(BIG*, SMALL*, UIDE)
C **
DEFINES THE STARTING POINT,
AND DISTANCE FROM THE AXIS FOR A PANEL.

COMMON/BLOKO/NAP*XAF~50)
~ZC~2~50~sCORD1~CORD2~CDlSIN~CD2SIN
COMMON/PHIRO/PHI(100),RO(100),SINE(100),COSINE(100)

READ (100) X1, Y1, Z1, CORD1
READ (101) X2, Y2, Z2, CORD2

Y1 = ABS(Y1)
Y2 = ABS(Y2)
READ (102) TVOR, TVNC, TVPDL
NVOR1 = INT(TVOR) + 1

IF (PDLoLEo360o) GO TO 1
READ (103) PHI1, ANG1, NAP
IF (PDLEo360o) GO TO 3
READ (104) PHI2, ANG2, NAP

IF (PDLoLEo360o) GO TO 1
READ (105) PHI3, ANG3, NAP
IF (PDLEo360o) GO TO 3
READ (106) PHI4, ANG4, NAP

C **
APPENDIX B. - Concluded

511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
APPENDIX C

SOURCE LISTING OF PROGRAM PLOTS

PROGRAM PLOTS(INPUT OUTPUT TAPE2 OUTPUT TAPE3 TAPE4 TAPE5)

C
THIS INITIALIZES THE PLOTTING Routines AND CALLS PLOTPAN.

CALL INITT(120)
CALL TERM(3,4096)
CALL CHRSIZ(4)
CALL PLOTPAN
CALL FINITI(0,0)
STCP
END

C

SUBROUTINE PLOTPAN

COMMON VORS(3,500)
COMMON /PRAMS/NVOR1,NVRNP0,D,RNCV,NVOR1,NVRN
COMMON /TITL/ TITLE(6)
COMMON /R3T/KARRAYZ(1),AMYTRIX(3,3)
INTEGER RNCV

C
THIS PROGRAM READS A SET OF 3 ANGLES AND PLOTS THE
CONFIGURATION AFTER ROTATING IT THROUGH THE INDICATED ANGLES.
FIRST ONE SIDE IS PLOTTED THEN IT IS REFLECTED THROUGH
ITS X-Z PLANE AND THE OTHER SIDE IS PLOTTED.
THE ROTATIONS ARE CARRIED OUT BY MATRIX MULTIPLICATION.
IT IS ALSO DETERMINED IF THE SUBPANELING AND CONTROL
POINTS ARE TO BE SHOWN.

LOGICAL SUBLINE,CPJNT2
DATA (KARRAY2(M),M=1,7)/20,3,3,500,3,3,3/
DATA PIE/3.14159/
REWIND 3
REWIND 4
REWIND 5

PLOTS 1
PLOTS 2
PLOTS 3
PLOTS 4
PLOTS 5
PLOTS 6
PLOTS 7
PLOTS 8
PLOTS 9
PLOTS 10
PLOTS 11
PLOTS 12
PLOTS 13
PLOTS 14
PLOTS 15
PLOTS 16
PLOTS 17
PLOTS 18
PLOTS 19
PLOTS 20
PLOTS 21
PLOTS 22
PLOTS 23
PLOTS 24
PLOTS 25
PLOTS 26
PLOTS 27
PLOTS 28
PLOTS 29
PLOTS 30
PLOTS 31
READ(3)NPAN,TITLE,TESTRTO

C HERE THE ANGLES OF ROTATION ARE READ; ROLL, PITCH, AND YAW,
C ARE READ.

CALL ERASE
WRITE(2,104)
READ *,ROLL
IF(ROLL.GT.360.)RETURN
WRITE(2,105)
READ *,PITCH
WRITE(2,106)
READ *,YAW
WRITE(2,107)
READ 112,SUBLINE
WRITE(2,108)
READ 112,CPONTZ
104 FORMAT(" INPUT THE ROLL ANGLE FOR THE AIRCRAFT",/,
2" DEG), >360 TO STOP.")
105 FORMAT("PITCH ANGLE")
106 FORMAT("YAW ANGLE")
107 FORMAT("SUB_PANELING ? TRUE OR FALSE")
108 FORMAT("CONTROL POINTS ? TRUE OR FALSE")
112 FORMAT(L7)

C IF(TESTRTO.LT.1.00)GO TO 500
IF(TESTRTO.GT.1.25)TESTRTO=1.25
RYY=5.0/TESTRTO
IXX2=INT(TESTRTO*3200)
IXX1=INT((4096-IXX2)*.5)
CALL SWINDO(IXX1,IXX2,1,3200)
CALL DWINDO(-5.0,5.0,-RYY,RYY)
GO TO 501

500 CALL SWINDO(450,3200,1,3200)
CALL DWINDO(-5.0,5.0,-5.0,5.0)

501 CALL TURNIT("ROLL,PITCH,YAW")
3 CALL MOVABS(100,3500)
CALL ANMODE

APPENDIX C. - Continued
APPENDIX C. - continued

WRITE(2,788) TITLE, ROLL, PITCH, YAW
788 FORMAT(8A10) /* THE ROLL ANGLE IS */,F7.1,
1/* THE PITCH ANGLE IS */,F7.1,
2/* THE YAW ANGLE IS */,F7.1
DO 300 NPANS=1,NPAN
DO 299 L=1,2
READ(3) NVROR, RNCR, PDL, IQUANT, NVRNPRD, ISETS
ISET=ISETS
NVROR=NVRNPRD-NVROR+1
IF(SUBLINE.A.PDL.LT.360)CALL VORSUB(5H LINES, ISETS, 3, L)
IF( .NOT. SUBLINE.A.PDL.LT.360)CALL VORSUB(5H LINES, ISETS, 3, L)
IF(SUBLINE.A.PDL.GT.360)CALL R4DLINE(L)
IF( .NOT. SUBLINE.A.PDL.GT.360)CALL VORSUB(5H EDGES, ISETS, 3, L)
IF(PDL.GF.+0.0.ACPOINTZ)CALL CPPOINTS(L)
IF(SUBLINE.A.PDL.GT.360.ISET=0)
IF(IL.EQ.2.OR.IQUANT.EQ.1) GO TO 300
CALL BACKUP(3+ISETS)
299 CONTINUE
300 IF(IL.EQ.2.A.ISET.EQ.0.OR.IQUANT.EQ.1.A.ISET.EQ.0) CALL SKPFILE(3, ISETS)
CALL MOVABS(100,3500)
CALL BIGER(III)
IF(III-1)9,9,3
END

SUBROUTINE VORSUB(WHICH, ISETS, IFILE, L)
COMMON /PRAMS/NVROR, NVRNPRD, RNCR, NVROR1, NVRN
COMMON VORS(3,500)
C
C THIS SUBROUTINE PLOTS A FLAT MAJOR PANEL WITH
C OR WITHOUT SUBPANELING, OR A CURVED PANEL WITHOUT SUBPANELING.
C
CALL SAVE(IFUG, IFILE, IDUA, 1, L, ISETS)
IF(WHICH.EQ.5H LINES)GO TO 1
N1=NVRN+1
C
PLOTS 70
PLOTS 71
PLOTS 72
PLOTS 73
PLOTS 74
PLOTS 75
PLOTS 76
PLOTS 77
PLOTS 78
PLOTS 79
PLOTS 80
PLOTS 81
PLOTS 82
PLOTS 83
PLOTS 84
PLOTS 85
PLOTS 86
PLOTS 87
PLOTS 88
PLOTS 89
PLOTS 90
PLOTS 91
PLOTS 92
PLOTS 93
PLOTS 94
PLOTS 95
PLOTS 96
PLOTS 97
PLOTS 98
PLOTS 99
PLOTS 100
PLOTS 101
PLOTS 102
PLOTS 103
PLOTS 104
PLOTS 105
SUBROUTINE REFLECT(N)

COMMON VORS(*,500)

THIS SUBROUTINE REFLECTS THE CONFIGURATION ACROSS ITS X-Z PLANE.

DO 1 J=1,N
1 VORS2,J) = -VORS2,J)

RETURN

END
SUBROUTINE BIGER(I)

THIS SUBROUTINE CAUSES A PORTION OF THE PLOT DEFINED WITH
THE GRAPHICS CURSOR, TO BE ENLARGED.

CALL ANMODE
WRITE(2, 103)
READ 102, ENLARGE
IF (ENLARGE.EQ.3) GO TO 1
I = 1
RETURN
1 CALL VCUPSR(ICHAR, X1, Y1)
CALL VCURSR(ICHAR, X2, Y2)
CALL FPAS1
RATI0 = (X2-X1)/(Y2-Y1)
IF (RATI0.GT.1.25) GO TO 10
IX1 = 4F
IX2 = 4F*7
IY2 = INT(4F*7*RATI0)
IY1 = INT((3200-IY2)*.5)
GO TO 2:
10 IY1 = 1
IY2 = 3200
IX2 = INT(3200*RATIO)
IX1 = INT((4098-IX2)*.5)
20 CALL SWINDO(IX1, IX2, IY1, IY2)
CALL DWINDO(IX1, IX2, IY1, IY2)
REWARD
READ(3)
I = 2
RETURN
102 FORMAT(1A3)
103 FORMAT(/'/, " FOR ENLARGEMENT INPUT YES")
END
SUBROUTINE CPOINTSTL

THIS SUBROUTINE PLOTS X'S ON THE CONTROL POINTS.

COMMON VORS(3:500), NVOR, NVORPRO, NVOR1, NVRN

IF(LIEEQ.21 CALL BACKUP(ISETS+1)

READ(C) NVORIRNCV, PRDIRNCV, NNNPNNNNPNVRNPRDINNO#

CALL SAVEIFUG(4, IDUHr ITL, ISETSI

CALL SAVE(IFUG, CINNIZILI ISETS)

DO 1 J=1, NNOINVOR

1 K=K+J-Z+4+ISETSG

CALL MOVEA(VORS(l+K*l) VORS(3*K+3)

CALL MOVEA(VORS(l+K*1) VORS(3*K+4)

CALL ORAUA(VORS(l+K*1) VORS(3*K+4)

IF(Kb.EAQ.500) CALL SAVE(IFUGP1, NNPVILI ISets)

RETURN

END
SUBROUTINE SAVE(FUG, FILE, P, TEST, L, SETS)
C THIS SUBROUTINE MAKES IT POSSIBLE TO REDUCE THE CORE
C REQUIREMENTS OF THE PROGRAM.
INTEGER FUG, FILE, P, TEST, SETS
COMMON VORS(3, 500)
COMMON /PRAMS/NVOR, NVRNPRD, RNCEV, NVOR1, NVRN
COMMON /ROE/KARRAY2(7), AMYTRIX(3, 3)
IF (TEST.NE.1) GO TO 1
FUG=-500*TEST2
IF (TEST2.EQ.1) RETURN
TEST2=TEST2+1
IF (TEST2.LT.SETS) N=500
IF (TEST2.EQ.SETS) N=MOD(P, 500)
IF (N.EQ.0) N=500
READ(FILE)((VORS(I, J), I=1, 3), J=1, N)
IF (L.EQ.2) CALL REFLECT(N)
KARRAY2(4)=N
CALL MATOPS(KARRAY2, AMYTRIX, VORS, VORS)
RETURN
1 TEST2=FUG*0
RETURN
END
C
SUBROUTINE BACKUP(I, N)
C CAUSES FILE I TO BE BACKSPACED N RECORDS.
DO 1 J=1, N
1 BACKSPACE I
RETURN
END
C
SUBROUTINE SKPFILE(I,N)
    CAUSES FILE I TO BE ADVANCED N RECORDS.
    DO 1 J=1,N
        1 READ(1)
    RETURN
END

SUBROUTINE TURNIT(RO,PO,YO)
    THIS SUBROUTINE DETERMINES THE MATRIX WHICH WILL ROTATE
    THE AIRCRAFT THROUGH THE ANGLES ROLL PITCH AND YAW AS
    THEY ARE NORMALLY DETERMINED.
    DIMENSION ZAX(3,3),XAX(3,3),R1(3,3),R2(3,3),VECTOR(3)
    DIMENSION KMULT(7),MULTV(7)
    COMMON /ROT/ <ARRAY2(7),AMYTRIX(3,3)
    DATA(KMULT(M),M=1,7) /20,6*3(/)
    DATA(MULTV(M),M=1,7) /20,3,3,1,3,3,3/
    DATA PIE/3.14159/
    DATA XAX(1,2),XAX(1,3),XAX(2,1),XAX(3,1)/4*(0,0)/
    DATA ZAX(1,3),ZAX(2,3),ZAX(3,3)/4*(0,0)/
    DATA R1(1,2),R1(1,3),R1(2,1),R1(3,1)/4*(0,0)/
    DATA R2(1,2),R2(2,1),R2(2,3),R2(3,2)/4*(0,0)/
    DATA ZAX(1,1),ZAX(2,3),R1(1,1),R2(2,2)/4*(1,0)/
    1 AMYTRIX(1,1)=AMYTRIX(3,3)=0
    AMYTRIX(1,2)*AMYTRIX(1,3)=AMYTRIX(2,1)=AMYTRIX(2,3)=0
    AMYTRIX(3,1)=AMYTRIX(3,2)=0
    VECTOR(1)=VECTOR(2)=0
    VECTOR(3)=0
    R=RO*PIE/180.
    P=-PO*PIE/180.
    Y=-YO*PIE/180.
    R0=PIE*5-R
    XAX(2,2)=XAX(3,3)=COS(R)
    SINE=SIN(R)
    XAX(2,3)=SINE
    XAX(3,2)=-SINE
ZAX(1,1) = ZAX(2,2) = COS(P)
SINE = SIN(P)
ZAX(1,2) = -SINE
ZAX(2,1) = SINE
R1(2,2) = R1(3,3) = COS(RTO)
SINE = SIN(RTO)
R1(2,3) = SINE
R1(3,2) = -SINE
CALL MATOPS(KMULT, XAX, AMYTRIX, AMYTRIX)
CALL MATOPS(KMULT, R1, AMYTRIX, AMYTRIX)
CALL MATOPS(KMULT, XAX, AMYTRIX, AMYTRIX)
TRANSPOSE MATR IX R1.
R1(2,3) = -SINE
R1(3,2) = SINE
CALL MATOPS(KMULT, R1, AMYTRIX, AMYTRIX)
5
ZAX(1,1) = ZAX(2,2) = COS(Y)
SINE = SIN(Y)
ZAX(1,2) = -SINE
ZAX(2,1) = SINE
CALL MATOPS(MULTV, AMYTRIX, VECTOR, VECTOR)
A = VECTOR(1)
B = VECTOR(2)
C = VECTOR(3)
V = SQRT(B*B + C*C)
BOVRV = B/V
R1(2,2) = R1(3,3) = C/V
R1(2,3) = -BOVRV
R1(3,2) = BOVRV
R2(1,1) = R2(3,3) = V
R2(1,3) = A
R2(3,1) = A
CALL MATOPS(KMULT, R1, AMYTRIX, AMYTRIX)
CALL MATOPS(KMULT, R2, AMYTRIX, AMYTRIX)
CALL MATOPS(KMULT, XAX, AMYTRIX, AMYTRIX)
TRANSPOSE MATR IX R1 AND MATR IX R2.
R1(2,3) = BOVRV
/GET,PLOTIT/UN=214737C
/CALL,PLOTIT(T=UORLAXX)
DO YOU WANT CAMBER IN THE PANELS?
TRUE OR FALSE
? TRUE

INPUT THE ROLL ANGLE FOR THE AIRCRAFT
(DEG), >360 TO STOP.
? 11
PITCH ANGLE
? 12
YAW ANGLE
? 13
SUBPANELING? TRUE OR FALSE
? TRUE
CONTROL POINTS? TRUE OR FALSE
? TRUE

Figure 2. - Sample execution of program PLOTIT.
(a) First location of graphics cursor.

Figure 4. - Use of graphics cursor to enlarge a section of a configuration plot.
(b) Second location of graphics cursor.

Figure 4. - Continued.